ELSEVIER International Journal of Approximate Reasoning 18 (1998) 35–52

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

Fusion of ordinal information using weighted median aggregation

Ronald R. Yager [1]

Machine Intelligence Institute, Iona College, New Rochelle, NY 10801-1890, USA

Received 1 August 1996; accepted 1 June 1997

Abstract

The weighted median is introduced as a fusion operation which can be used in situations in which, while having numeric values for the weights associated with the objects to be fused, the actual objects being fused only satisfy an ordering property. After introducing the concept of weighted median we compare it with the weighted average and show that they have many properties in common. We then provide an algorithm for learning the weights associated with a median aggregation. We then show how we can use this technique to extend the applicability of the Ordered Weighted Averaging (OWA) operator to situations in which the arguments are nonnumeric. Finally we show how we can use the weighted median as an alternative to the expected value in the evaluation of probabilistic lotteries. © 1998 Elsevier Science Inc.

1. Introduction

Many kinds of information fusion techniques are based upon the use of the averaging operator. One fundamental requirement for using this operator is that the objects to be aggregated are numeric. As technological interest moves to the construction of intelligent systems, we are often faced with problems in which we must fuse nonnumeric information. The median provides an aggregation operation which only requires that the fused objects can be ordered. Thus while the median aggregation, like the average, can work in numeric do-

[1] Tel.: +1-212-249-2047; fax: +1-212-249-1689; e-mail: yager@panix.com.

PII S0888-613X(97)10003-2

mains, it also can be used in situations in which we are not interested in combining numbers but ordered objects. One important situation in which this occurs is when we have linguistic information to be fused. In this work we consider a new fusion operator which is an extension of the median. This operator is called the weighted median and was originally introduced in [1,2]. With this operator we can provide a fusion of weighted objects where the objects to be aggregated need only be drawn from an ordered set while the weights can be numbers. A prototypical example of this situation occurs in decision making under risk. In this environment we could have probabilities associated with the state of nature but the payoffs could only be evaluated in terms such as *bad*, *good* and *excellent*. Actually at a deeper level the ability to aggregate weighted ordinal information allows us to put less burden on the providers of information in that they only need give preference information in terms of an ordering, this is a consideration of some importance in applications where there are often problems getting the users of decision making tools to provide the required data. After introducing the concept of weighted median we compare it with the weighted average and show that they have many properties in common. We then provide an algorithm for learning the weights associated with a median aggregation. We then show how we can use this technique to extend the applicability of the Ordered Weighted Averaging (OWA) operator [3] to situations in which the arguments are nonnumeric. Finally we show how we can use the weighted median as an alternative to the expected value in situations where the values are nonnumeric.

2. Weighted median aggregation

An often used approach to aggregation of scores is the weighted average. We recall that if $a_1, a_2, \ldots, a_n$ are a collection of values and $w_1, w_2, \ldots, w_n$ are a collection of associated weights where it is assumed that:

Property 1. $\sum w_i = 1$,

Property 2. $w_i \in [0, 1]$, $\forall i$,

then the weighted average is

$$\bar{x} = \sum_{i=1}^{n} w_i a_i.$$

In environments in which the scores to be aggregated are nonnumerical this method cannot be used. In the following, we shall consider an alternative approach to aggregation which is useful in environments in which the scores to be aggregated are drawn from a scale which only has a linear ordering, while the weights are still numeric values satisfying properties one and two above. This approach is called the *weighted median* [1,2].

In the following we shall let $L = \{r_1, r_2, \ldots, r_q\}$ be a set having the property $r_i > r_j$, if $i > j$. Specifically we assume the elements in L are ordered. A particular case of this type of situation is when the set of elements are of a linguistic type such as

$$\{\text{very low, low, medium, high, very high}\}.$$

Assume D is a collection (multi-set) of elements drawn from L. One approach to aggregating these scores is to use the median operator. We recall that if $D = \{a_1, a_2, \ldots, a_n\}$, then if b_j is the jth largest of the elements in D

$$\mathrm{Med}(D) = \begin{cases} b_{\left(\frac{n+1}{2}\right)} & \text{if } n \text{ is odd,} \\ b_{\left(\frac{n}{2}\right)} & \text{if } n \text{ is even.}^2 \end{cases}$$

Example 1. Let $D = \{\text{med, low, high, very high, high, low, very high}\}$. Thus in this case $b_1 = \text{very high}$; $b_2 = \text{very high}$; $b_3 = \text{high}$; $b_4 = \text{high}$; $b_5 = \text{med}$; $b_6 = \text{low}$; $b_7 = \text{low}$. Since $n = 7$ then $\mathrm{Med}(D) = b_4 = \text{high}$.

In [2] Yager and Rybalov considered the problem of weighted median aggregation. Assume $D = \langle (w_1, a_1), (w_2, a_2), \ldots, (w_n, a_n) \rangle$ are a collection of pairs where a_i is a score and w_i is its associated weight. We again assume that the weights satisfy Properties 1 and 2. Again assume that the a_i are reordered such that b_j is the jth largest of the a_i. Furthermore, let u_j be the weight that is associated with the a_i that becomes b_j. Thus if $b_j = a_5$ then $u_j = w_5$. Once having the ordered collection

$$\hat{D} = \langle (u_1, b_1), (u_2, b_2), \ldots, (u_n, b_n) \rangle$$

to calculate the weighted median, we proceed as follows. We denote

$$T_i = \sum_{j=1}^{i} u_j,$$

the sum of the first i weights. From this we get

$$\text{Weight Med}(D) = b_k,$$

where k is such that

$$T_{k-1} < 0.5 \quad \text{and} \quad T_k \geqslant 0.5.$$

Thus the weighted median is the ordered value of the arguments for which the sum of the weights first crosses the value of 0.5. We shall call k the cross over value.

---

[2] In using the median it should be pointed out that when n is even the value of the median is not unique. More generally $\mathrm{Med}(D) \in [b(n/2),\ b(n/2+1)]$. Usually some convention is adopted for the selection, for example taking the largest value of the range as we have indicated.

The following example illustrates this type of aggregation.

Example 2. Assume our scale for obtaining values is

$L = \{\text{very low (vl), low (l), medium (m), high (h), very high (vh)}\}.$

Assume

$D = \langle(0.3, \text{l}), (0.2, \text{h}), (0.2, \text{m}), (0.1, \text{vl}), (0.2, \text{vh})\rangle.$

Ordering the arguments we get

b_j	u_j	T_j	
vh	0.2	0.2	
h	0.2	0.4	
m	0.2	0.6	←
l	0.3	0.9	
vl	0.1	1.0	

Hence Weight Med(D) = medium.

In the preceding example we have introduced an aggregation technique which only requires that the values to be aggregated are drawn from some ordered set while allowing the weights associated with these elements to be numbers.

While we assumed that the weights associated with the elements to be aggregated are drawn from the unit interval and sum to one this is not necessary. If the weights are assumed to be nonnegative numbers we can normalize these values and then proceed to use the weighted median.

In Section 3 we shall look at the properties of the weighted median (WM) and compare it to the weighted average (WA).

3. Properties of weighted median

Let us now look at some of the properties of this weighted median and compare them with the corresponding properties of the weighted average.

We first note that both are idempotent. Assume $a_i = a$ for all i. In this case

$$\text{WA} = \sum_{i=1}^{n} w_i a_i = \sum_{i=1}^{n} w_i a = a \sum_{i=1}^{n} w_i = a.$$

In the case of the weighted median if $a_i = a$ for all i then $b_j = a$ for all j. Thus if k is the crossover value, since $b_j = a$ for j then $b_k = a$.

We also note that the weighted average is commutative (generally symmetric) in the sense that each of the pairs (w_i, a_i) are treated in the same manner. It can be easily seen that each of the pairs (w_i, a_i) are also treated in the same manner in the weighted median.

We next consider the property of monotonicity. Let

$$D = \langle(w_i, a_i)\rangle \quad i = 1, \dots, n,$$
$$\hat{D} = \langle(w_i, \hat{a}_i)\rangle \quad i = 1, \dots, n,$$

where for each i we have $\hat{a}_i \geqslant a_i$. Monotonicity requires that the aggregation of $\hat{D}$ be at least as great as the aggregation of D. In the case of the weighted average

$$\mathrm{WA}(D) = \sum_{i=1}^{n} w_i a_i, \qquad \mathrm{WA}(\hat{D}) = \sum_{i=1}^{n} w_i \hat{a}_i$$

since $\hat{a}_i \geqslant a_i$ then

$$\mathrm{WA}(\hat{D}) \geqslant \mathrm{WA}(D).$$

To show the monotonicity of the weighted median is slightly more complex. First we note that to show the monotonicity property for the weighted median all we need show is that if one of the values in the argument increases and all others remain the same, then the aggregation cannot decrease. Thus in the following we shall assume $D = \langle(w_i, a_i)\rangle$ for $i = 1, \dots, n$ and $\hat{D} = \langle(w_i, \hat{a}_i)\rangle$ where for some index m, $\hat{a}_m > a_m$ and $\hat{a}_i = a_i$ for $i \neq m$.

For simplicity we shall assume that the a_i's have already been indexed so that $a_i \geqslant a_j$ if $i < j$. In the following we shall assume that the crossover for $\mathrm{WM}(D)$ occurs at k, hence $T_{k-1} = \sum_{j=1}^{k-1} w_j < 0.5$ and $T_k = \sum_{j=1}^{k} w_i \geqslant 0.5$ and thus $\mathrm{WM}(D) = a_k$. To show the monotonicity of $\hat{D}$ we need consider three cases regarding the location m with respect to k.

(1) $m < k$: In this case we still have $T_{k-1} < 0.5$ and $T_k \geqslant 0.5$ and thus

$$\mathrm{WM}(\hat{D}) = a_k = \mathrm{WM}(D)$$

since a_k is unchanged.

(2) $m = k$: In this situation it is still the case that

$$T_k = \sum_{j=1}^{k} w_i \geqslant 0.5$$

and hence

$$\mathrm{WM}(\hat{D}) \in \{a_1, a_2, \dots, a_{k-1}, \hat{a}_m\}.$$

Since all these values are at least as great as a_k it follows that $\mathrm{WM}(\hat{D}) \geqslant \mathrm{WM}(D)$.

(3) $m > k$: Two cases must be considered. (i) $\hat{a}_m \leqslant a_k$: In this case the reordering of the elements leads us to $b_i = a_i$ for $i \leqslant k$ and hence the weighted median still occurs at a_k. (ii) $\hat{a}_m > a_k$: In this case it follows that

$$\sum_{j=1}^{k} w_i + w_m \geqslant 0.5$$

and hence

$$\mathrm{WM}(\hat{D}) \in \{a_1, \ldots, a_k, \hat{a}_m\}.$$

Since all these values are at least as great as a_k then

$$\mathrm{WM}(\hat{D}) \geqslant \mathrm{WM}(D).$$

The satisfaction of the above three conditions implies that the weighted median is a mean operator [4,5].

We now show that the introduction of an element with zero weight does not effect the aggregation. Let

$$D = \langle (w_1, a_1), (w_2, a_2), \ldots, (w_n, a_n) \rangle$$
$$\hat{D} = \langle (w_1, a_1), (w_2, a_2), \ldots, (w_n, a_n), (0, a_{n+1}) \rangle.$$

For the weighted mean we have

$$\mathrm{WM}(D) = \sum_{j=1}^{n} w_i a_i,$$

$$\mathrm{WM}(\hat{D}) = \sum_{j=1}^{n+1} w_i a_i = \sum_{j=1}^{n} w_i a_i + 0 \cdot a_i = \mathrm{WM}(D).$$

Let us now consider the weighted median, without loss of generality we shall assume that for $i = 1, \ldots, n$ the a_i's are indexed so that $a_i \geqslant a_j$ for $i < j$. Assume that $\mathrm{WM}(D) = a_k$ that is

$$\sum_{j=1}^{k-1} w_j < 0.5 \quad \text{and} \quad \sum_{j=1}^{k} w_j \geqslant 0.5.$$

Let us now consider $\hat{D}$. First, assume that $a_{n+1} \leqslant a_k$. In this case we see that the introduction of $(0, a_{n+1})$ will not effect the crossover point. Now assume $a_{n+1} > a_k$. Since $w_{n+1} = 0$ it still does not effect the crossover point, because we must include the original first k values to get the total up to 0.5.

We now show that the occurrence of two elements with the same value can be simplified into one element by just adding the weights associated with the element having equal value. Let

$$D = \langle (w_1, a_1), (w_2, a_2), \ldots, (w_n, a_n), (w_{n+1}, a_n) \rangle,$$
$$\hat{D} = \langle (w_1, a_1), (w_2, a_2), \ldots, (w_n + w_{n+1}, a_n) \rangle.$$

The fact that

$$\mathrm{WA}(D) = \mathrm{WA}(\hat{D})$$

is obvious from the definition of the WA.

Consider now the case of weighted median. In the case of D we order the values and obtain

$$(b_i, u_i), \quad i = 1, \ldots n + 1,$$

where $b_i \geqslant b_j$ if $i < j$. Because of the fact that $a_n = a_{n+1}$ they will be adjacent to each other in the ordering. Thus if a_n becomes b_m then a_{n+1} becomes b_{m+1}. Recall that the weighted median is the value b_k such that $\sum_{j=1}^{k-1} u_j < 0.5$ and $\sum_{j=1}^{k} u_j \geqslant 0.5$. In the case of $\hat{D}$ the ordering will be the same except the element b_{m+1} will be eliminated. If $k \leqslant m$ then the determination of WM$(\hat{D})$ will be unaffected by the fact that we have replaced the pair by one element. If $k = m + 1$, then because of the change we get that $\hat{k}$ occurs at $\hat{k} = m$, however, since $b_m = a_{n+1}$ we get the same result. If $k > m + 1$ then the result again is unaffected by this change.

We now consider the situation is which all the weights associated with the values are the same. We show that this situation leads to respectively the arithmetic average and the ordinary median aggregation. Assume

$$D = \left\langle \left(\frac{1}{n}, a_i\right)\right\rangle, \quad i = 1, \ldots, n.$$

In the case of the weighted average we get the arithmetic average

$$\mathrm{WA}(D) = \frac{1}{n}\sum_{i=1}^{n} a_i.$$

We turn to the weighted median, for simplicity we shall assume that the a_i's have been indexed in descending order, $a_i \geqslant a_j$ where $i < j$. To obtain the weighted median we calculate k such that

$$T_{k-1} = \sum_{j=1}^{k-1} \frac{1}{n} < 0.5 \quad \text{and} \quad T_k = \sum_{j=1}^{k} \frac{1}{n} \geqslant 0.5$$

and therefore we require:

$$T_{k-1} = \frac{k-1}{n} < 0.5, \qquad T_k = \frac{k}{n} \geqslant 0.5.$$

Let us first assume that n is odd and let $k = (n + 1)/2$. We see that;

$$T_{k-1} = \left(\frac{n+1}{2} - 1\right)\frac{1}{n} = \left(\frac{n-1}{2}\right)\frac{1}{n} = \frac{n}{2n} - \frac{1}{2n} = \frac{1}{2} - \frac{1}{2n} < 0.5,$$

$$T_k = \left(\frac{n+1}{2}\right)\frac{1}{n} = \frac{n+1}{2n} = \frac{1}{2} + \frac{1}{2n} \geqslant 0.5.$$

Thus the weighted median occurs at a_k where $k = (n + 1)/2$, which is the ordinary median of the arguments. Assume n is even. Consider $k = n/2$. In this case

$$T_{k-1} = \left(\frac{n}{2} - 1\right)\frac{1}{n} = \left(\frac{n-2}{2}\right)\frac{1}{n} = \frac{1}{2} - \frac{1}{n} < 0.5,$$

$$T_k = \left(\frac{n}{2}\right)\frac{1}{n} = \frac{1}{2}.$$

Thus the weighted median occurs at a_k where $k = n/2$ which is again the median.

Thus we see that the case when all the weights are equal $1/n$ the weighted median gives us the ordinary median.

We describe another property shared by the weighted average and weighted median. Specifically, we show that in both these aggregations if weight moves from lower values to higher values the aggregated value cannot decrease. Without loss of generality we shall assume in the following the a_i's are indexed such that $a_i \geqslant a_j$ for $i < j$. Consider the arguments

$$D = \langle (w_1, a_1), (w_2, a_2), \ldots, (w_n, a_n) \rangle,$$
$$\hat{D} = \langle (\hat{w}_1, a_1), (\hat{w}_2, a_2), \ldots, (\hat{w}_n, a_n) \rangle.$$

Let the weights $\hat{w}_i$ and w_i be related as follows

$$\hat{w}_m = w_m - \Delta \quad (\Delta < w_m),$$
$$\hat{w}_q = w_q + \Delta,$$
$$\hat{w}_i = w_i \quad \text{for all other } i$$

and $q < m$.

Thus we have assumed some amount of the weight from w_m is moved to w_q.

If we calculate

$$\mathrm{WA}(D) = \sum_{j=1}^{n} w_i a_i,$$
$$\mathrm{WA}(\hat{D}) = \sum_{j=1}^{n} \hat{w}_i a_i = \sum_{j=1}^{n} w_i a_i + \Delta a_q - \Delta a_m$$

and therefore we see that

$$\mathrm{WA}(\hat{D}) - \mathrm{WA}(D) = \Delta a_q - \Delta a_m.$$

Since it is assumed $a_q \geqslant a_m$ we get

$$\mathrm{WA}(\hat{D}) - \mathrm{WA}(D) \geqslant 0.$$

We now consider the case of weighted median for the case of D, we get

$$\mathrm{WM}(D) = a_k,$$

where

$$T_{k-1} = \sum_{j=1}^{k-1} w_j < 0.5,$$
$$T_k \geqslant 0.5.$$

For the case of $\hat{D}$, we get

$$\mathrm{WM}(\hat{D}) = a_{\hat{k}},$$

where

$$T_{\hat{k}-1} = \sum_{j=1}^{\hat{k}-1} \hat{w}_j < 0.5,$$

$$T_{\hat{k}} = \sum_{j=1}^{\hat{k}} \hat{w}_j \geqslant 0.5.$$

Since under our assumption we have moved weights from element m to q and $q < m$ then it follows that $\hat{k} \leqslant k$ and then since the a_i's are indexed in descending order we have

$$a_{\hat{k}} \geqslant a_k$$

and thus

$$\mathrm{WM}(\hat{D}) \geqslant \mathrm{WM}(D).$$

In Section 2 we have shown that the weighted median and weighted average share a number of fundamental properties. However, while the weighted average requires numeric values for objects to be aggregated the weighted median only requires the objects to be aggregated be drawn from a ordinal scale. These observations leads one to consider the weighted median as an alternative to the weighted average in environments in which the objects to be aggregated are ordinal values.

4. Learning weights in weighted median aggregation

In Section 3 we have shown that the weighted median provides a aggregation technique similar in spirit to weighted average. We have particularly noted its usefulness in environments in which the values to be aggregated are drawn from ordinal scales while the weights associated with these values are numeric.

In this section we consider the problem of learning the weights in the weighted median aggregation from observations on data. The technique we develop is very similar in spirit to the type of gradient techniques used in finding the weights associated with weighted average aggregation [6].

Assume $V_1, V_2, \ldots, V_n$ are a collection of variables which take their values in the space

$$L = \{r_1, r_2, \ldots, r_q\}.$$

We further assume that the space L has only an ordering relation, that is

$$r_i > r_j \quad \text{if } i < j.$$

As we noted earlier the elements of L can be associated, for example, with a set of linguistic labels.

Assume we have a collection of observations $A_1, A_2, \ldots, A_p$. Specifically each A_i is a vector

$$A_i = \begin{bmatrix} a_{i1} \\ a_{i2} \\ \vdots \\ a_{in} \end{bmatrix},$$

where a_{ij} is an element in the space L corresponding to the value of V_j in the ith observation. Associated with each vector observation A_i is a value $y_i \in L$ indicating the associated aggregated value for this observation. Thus we have a collection of pairs (A_i, y_i) $i = 1, \ldots p$ corresponding to our data.

A prototypical example of this situation is the case in which each V_j is a criteria in a multiple criteria decision problem. For any possible alternative solution x_i we can consider A_i as corresponding to the satisfaction of the criteria by this alternative. Thus a_{ij} is the degree of satisfaction of the ith alternative to the jth criteria. In this framework y_i is the overall evaluation of this alternative to the collection of criteria.

The problem of interest here is in finding a collection of normalized weights, w_i, $i = 1, \ldots, n$ such that the weighted median aggregation

$$\mathrm{WM}((w_1, V_1), (w_2, V_2), \ldots, (w_n, V_n)) = y$$

best matches our observed data.

Before proceeding we make some observations about distance on an ordinal scale. Assume we have a collection of values R drawn from the ordinal scale L. Let y be another element from L. We consider the problem of finding the element e in R that is the minimal distance from y.

1. If $y \in R$ then $e = y$.
2. If $y < \min_{x \in R} x$ then $e = \min_{x \in R} x$.
3. If $y > \max_{x \in R} x$ then $e = \max_{x \in R} x$.
4. For the remaining case we find two values e_1 and e_2 in R such that
   e_1 is the largest element in R less than y
   e_2 is the smallest element in R greater than y.
   In this case $e = \{e_1, e_2\}$.

The algorithm we propose is in the spirit of the gradient techniques used in neural networks. In particular each observation causes us to modify our estimate of the weights. The updation algorithm is continuously used until we pass through the data collection with minimal change in the weights. In the following we describe the learning or updation algorithm.

Assume $(w_1, w_2, \ldots, w_n)$ are our current estimates of the weights. Let $A = (a_1, a_2, \ldots, a_n)$ be our observed set of values for the V_i's. Without loss of generality we shall assume that the indexing has been done so that $a_i \geqslant a_j$ if $i < j$. Finally let y be the associated aggregated value for this data.

Our problem is to now update the weights based on this data.

Updation Algorithm

1. Calculate i^* s.t.

$$T_{i^*-1} = \sum_{i=1}^{i^*-1} w_i < 0.5,$$

$$T_{i^*} = \sum_{i=1}^{i^*} w_i \geqslant 0.5.$$

Thus a_{i^*} is the aggregated value the model provides based upon the current observation and current estimate of the weights.

2. Calculate the value in A that is closest to y, this is either a single value e or a set $\{e_1, e_2\}$.

3. If $a_{i^*} = e$ or $a_{i^*} \in \{e_1, e_2\}$ stop and no modification is necessary.

4. If the closest element is e denote this as $a_{\hat{i}} = e$. If the closest element is the set $\{e_1, e_2\}$ then set

$$(i) a_{\hat{i}} = e_1 \text{ if } a_{i^*} < e_1,$$

$$(ii) a_{\hat{i}} = e_2 \text{ if } a_{i^*} > e_2.$$

5. (i) If $T_{\hat{i}} < 0.5$ then set $\Delta = 0.5 - T_{\hat{i}}$ and update the weights as follows:

$$w_i' = \begin{cases} w_i\left(1 + \frac{\alpha\Delta}{T_{\hat{i}}}\right) & \text{for } i = 1, 2, \ldots, \hat{i}, \\ w_i\left(1 - \frac{\alpha\Delta}{1 - T_{\hat{i}}}\right) & \text{for } i = \hat{i} + 1, \ldots, n. \end{cases}$$

(ii) If $T_{\hat{i}-1} \geqslant 0.5$ then set $\Delta = T_{\hat{i}-1} - 0.5$ and update the weights as follows

$$w_i' = \begin{cases} w_i\left(1 - \frac{\alpha\Delta}{T_{\hat{i}-1}}\right) & \text{for } i = 1, 2, \ldots, \hat{i} - 1, \\ w_i\left(1 + \frac{\alpha\Delta}{1 - T_{\hat{i}-1}}\right) & \text{for } i = \hat{i}, \ldots, n. \end{cases}$$

This algorithm essentially adjusts the weights in a manner that tries to bring the model solution nearer the input value that is closest to the observed value. In the above α controls the learning rate and is a value in the unit interval. The selection of α is guided by the same principles used in the selection of the learning rate in most gradient ascent techniques such as back propagation, the larger α the more responsive we are to new observations.

Example 3. Assume

$$L = \{r_1, r_2, r_3, \ldots, r_n\}.$$

Let $a_1 = r_2, a_2 = r_3, a_3 = r_4, a_4 = r_6, a_5 = r_8$ and assume $y = r_6$ and let the current weights be

$$w_1 = 0.2, w_2 = 0.2, w_3 = 0.15, w_4 = 0.3, w_5 = 0.15.$$

Let $\alpha = 0.3$.

Step 1: $T_1 = 0.2, T_2 = 0.4, T_3 = 0.55, T_4 = 0.85, T_5 = 1$. Hence $i^* = 3$ and $a_{i^*} = r_4$.

Step 2: The value closest to y is $e = a_4 = r_6$.

Step 3: $a_{i^*} \neq r_6$.

Step 4: $\hat{i} = 4$.

Step 5: Since $T_{\hat{i}-1} = T_3 \geqslant 0.5$ then $\Delta = 0.55 - 0.5 = 0.05$ and

$$w_i' = w_i\left(1 - \frac{(0.3)(0.05)}{0.55}\right) = w_i(1 - 0.027) = (0.973)w_i, \quad i = 1,\ 2,\ 3,$$

$$w_i' = w_i\left(1 + \frac{(0.3)(0.05)}{0.45}\right) = w_i(1 + 0.033) = (1.033)w_i, \quad i = 4,\ 5.$$

From this we get

$$w_1 = 0.195, \quad w_2 = 0.195, \quad w_3 = 0.145, \quad w_4 = 0.31, \quad w_5 = 0.155.$$

5. Ordinal OWA aggregation

In [3] Yager introduced the OWA aggregation operators. We first recall this operator.

Definition 4. An OWA operator of dimension n is a mapping

$$F_W : \mathbb{R}^n \to \mathbb{R}$$

that has associated with it a weighting vector W of dimension n such that

$$w_i \in [0, 1] \quad \text{and} \quad \sum_{i=1}^{n} w_i = 1$$

and where the aggregated value is

$$F_W(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j b_j,$$

where b_j is the jth largest of the a_i.

In [3] it is noted that the OWA operator provides a family of mean aggregation operators. In particular, by selecting an appropriate weighting vector W we obtain a specific type of mean aggregation. One semantics that can associated with this operator is as a generalization of the arithmetic average. In particular, while the arithmetic average treats all the objects to be aggregated in the same manner, the OWA operator, via its ordering process, allows us to emphasize different arguments according to their position in this ordering. Thus by appropriately selecting the vector W we can, for example, put more weights on the higher scores or lower scores or middle scores. It should be noted that this is different from the weighted average which can also be seen as a generalization of the arithmetic average. While the weighted average also puts different emphasis on the arguments, this difference is based on some given fixed weight associated with that argument.

A number of special cases of this operator can be pointed out. If W is such that

$$w_k = 1, \quad w_i = 0 \quad \text{for } i \neq k,$$

then

$$F_w(a_1, a_2, \ldots, a_n) = b_k$$

that is we obtain the kth largest of the arguments. We note that if $k = 1$ then

$$F_w(a_1, a_2, \ldots, a_n) = \max_i [a_i]$$

and if $k = n$ then

$$F_w(a_1, a_2, \ldots, a_n) = \min_i [a_i].$$

If n is odd and $k = (n + 1)/2$ then

$$F_w(a_1, a_2, \ldots, a_n) = \mathrm{Med}[a_i].$$

Another special case in the one in which $w_i = 1/n$ for all i. In this case

$$F_w(a_1, a_2, \ldots, a_n) = \frac{1}{n} \sum_{i=1}^{n} a_i,$$

which is the simple average.

More generally, if the weights are located near the top of the weighting vector we tend to emphasize the higher values in the aggregation process while if the weights are located near the bottom emphasizes is put on the lower scores.

We now consider an extension of this operator to the case where the domain of the arguments is an ordinal set L, rather than being the real line. Thus we want

$$F_w : L^n \to L.$$

Thus we must calculate $F_W(a_1, a_2, \ldots, a_n)$ where $a_i \in L$. The original OWA operator essentially consists of two steps:

1. Order the a_i to obtain the b_j.
2. Calculate

$$\sum_{j=1}^{n} w_j b_j.$$

If the arguments are drawn from an ordered set L we see that while we still can perform step 1 we are now unable to implement step 2.

In order to extend the OWA operator to this ordinal domain we must find some appropriate operation to replace the weighted average used in step 2. Based upon our previous observations a natural choice to replace the weighted average used in step 2 is the weighted median. Thus if $a_i \in L$ then

$$F_W(a_1, a_2, \dots, a_n) = \text{Weight-Median } ((w_j, b_j)),$$

where b_j is the jth largest of the a_i.

Example 5. Let $L = \{l_1, l_2, \dots, l_q\}$ be any ordered set such that $l_i > l_j$ if $i < j$. Let

$$W = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.15 \\ 0.1 \\ 0.15 \end{bmatrix}.$$

Assume we desire to calculate $F_w(l_3, l_5, l_2, l_1, l_6)$. In this case $b_1 = l_1$, $b_2 = l_2, b_3 = l_3, b_4 = l_5, b_5 = l_6$ and $w_1 = 0.3, w_2 = 0.3, w_3 = 0.15, w_4 = 0.1$, $w_5 = 0.15$. If we define

$$T_i = \sum_{j=1}^{i} w_i,$$

then $T_1 = 0.3, T_2 = 0.6, T_3 = 0.75, T_4 = 0.85, T_5 = 1$ and we see that $i^* = .2$ and hence

$$F_w(l_3, l_5, l_2, l_1, l_6) = l_2.$$

Then the process of calculating the OWA aggregation of ordinal values can be very simply expressed. Let W be the weighting vector and let

$$T_i = \sum_{j-1}^{i} w_i,$$

then if i^* is the value such that

$$T_{i^*-1} < 0.5, \quad T_{i^*} \geqslant 0.5,$$

we get

$$F_w(a_1, a_2, \dots, a_n) = b_{i^*},$$

where b_{i^*} is the i^* largest of the a_j. Thus all we have to do is add up the weights, find the index of the weights that makes this sum cross over 0.5 and that ordered element is the OWA aggregation.

It is interesting to consider the special cases we introduced before. First consider the situation where w is such that $w_k = 1$ and $w_i = 0$ for $i \neq k$. In this case we see that $T_i = 0$ for $i < k$ and $T_i = 1$ for $i = k$, hence $i^* = k$ and we obtain, as in the case of the ordinary OWA operator, the kth largest argument.

Consider now the case of the simple average. In this case $w_i = 1/n$. We then calculate

$$T_i = \sum_{j=1}^{i} w_i = \frac{1}{n}\sum_{j=1}^{i} = \frac{i}{n}.$$

In this case crossover occurs when $i^*/n = 1/2$ and thus $i^* = n/2$. Thus if n is even $i^* = n/2$ and if n is odd $i^* = (n+1)/2$. Thus we see that the simple average becomes the median of the arguments.

We now consider a further generalization of the ordinal aggregation. Assume that we have an OWA weighting vector W of dimension n. Furthermore, assume we have a collection of arguments a_i. However, each of these arguments has an associated importance value which we shall indicate as u_i. Thus in this case we are interested in finding

$$F_w((u_i, a_i)).$$

For simplicity we shall assume that these important weights have been normalized so that

$$u_i \in [0, 1], \qquad \sum u_i = 1$$

also for simplicity we shall assume the a_i's have already been indexed in descending order, thus

$$a_i \geqslant a_j, \quad i < j.$$

In the following we shall let

$$T_i = \sum_{j=1}^{i} w_j, \quad S_i = \sum_{j=1}^{i} u_j.$$

The following procedure can be used to obtain $F_w((u_i, a_i))$:

1. Calculate the crossover point of the w_i. That is calculate i^* such that

$$T_{i^*} \geqslant 0.5, \quad T_{i^*-1} < 0.5.$$

2. Denote

$$\alpha = \frac{i^*}{n}.$$

3. Calculate the crossover point of the u_i. That is calculate k^* such that

$$S_{k^*} \geqslant \alpha, \quad S_{k^*-1} < \alpha.$$

4. $F_w((u_i, a_i)) = a_{k^*}$.

Example 6. Assume

$$W = \begin{bmatrix} 0.2 \\ 0.1 \\ 0.4 \\ 0.3 \end{bmatrix}.$$

Aggregate

$$((0.4, l_2), (0.1, l_4), (0.15, l_5), (0.35, l_1)).$$

In this case $n = 4$ and $i^* = 3$ hence $\alpha = 3/4$. Ordering the argument we get

b_i	u_i	s_i	
l_1	0.35	0.35	
l_2	0.4	0.75	←
l_4	0.1	0.85	
l_5	0.15	1.0	

Hence the aggregated value is l_2.

We now show that in the case when all the u_i are equal this reduces to the unweighted ordinal OWA aggregation. In the unweighted case if i^* is the crossover index for the OWA weights then the aggregated value is a_{i^*}. Consider now the weighted case. If all u_i are equal then $u_i = 1/n$. Furthermore we note that $\alpha = i^*/n$ where i^* is the crossover index for the OWA weights. Since

$$S_i = \sum_{j=1}^{i} u_i = \sum_{j=1}^{i} \frac{l}{n} = \frac{i}{n}$$

and when $k^* = i^*, S_i = i^*/n = \alpha$ we also get a_{i^*} as our aggregated value.

We can see that the proposed method is a generalization of the weighted median we introduced at the beginning. In particular we recall weighted median of $((u_i, a_i))$ requires us to order the values and then find the position where the weights crossover 0.5. In the case of the OWA aggregation with weights we have adjusted, based upon the OWA weighting vector, the crossover point.

6. Comparison of lotteries

In some decision making environments, decision making under risk, the selection of an alternative rather than leading to a deterministic payoff may result in a nondeterministic outcome called a lottery. A lottery is defined as a situation in which we have a set of outcomes each associated with a probability (see Fig. 1).

In Fig. 1 $a_1, a_2, \ldots, a_n$ are the outcomes and p_i is the probability associated with outcome a_i. We note of course $p_i \in [0, 1]$ and $\sum_{i=1}^{n} p_i = 1$. We shall formally express this lottery as

$$\text{Lot}\ ((p_i, a_i)).$$

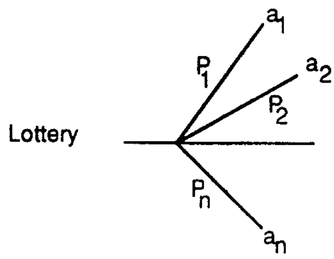

Fig. 1. Lottery.

In trying to choose between alternative actions whose results are lotteries we are faced with the problem of comparing lotteries. Specifically assume we have two lotteries

Lottery 1 Lot $((p_i, a_i))$,

Lottery 2 Lot $((\hat{p}_i, b_i))$.

We are interested in determining which is the preferred lottery.

The usual way of adjudicating this problem is to calculate the expected value of each of the lotteries

$$EV(L_1) = \sum_{i=1}^{n} a_i p_i,$$

$$EV(L_2) = \sum_{i=1}^{n} b_j \hat{p}_i$$

and select as the preferred alternative the one with the larger expected value.

In environments in which the outcomes are not numbers we cannot use the above procedure. If an ordinal ranking can be assigned to the outputs, we can use the weighted median aggregation. Thus if the a_i's and b_i's are drawn from the same ordinal scale L we can calculate the weighted median of each of lotteries.

$$\text{Value}(L_1) = \text{Weighted Median } ((a_i, p_i)),$$

$$\text{Value}(L_2) = \text{Weighted Median } ((b_i, \hat{p}_i)).$$

We select as the preferred alternative the one which has the largest weighted median associated with its lottery.

7. Conclusion

We introduced a fusion operator called the weighted median and investigated its properties. We showed that this operator shares many of the properties of the ordinary weighted average while only requiring that the values to be aggregated are drawn from a linear scale. This capability makes it a potentially useful tool for operations that require the mixing of numeric and linguistic values and as such it can play a central role in the development of the *computing with words* paradigm introduced by Zadeh [7]. We showed how we can use the weighted median operator as an alternative to the expected value in the evaluation of probabilistic lotteries in cases in which the payoffs are nonnumeric. Another potential application of this operator is in fuzzy logic control where the controlled variable is discrete, such as the setting of a dial [8]. In these types of problems the algorithm we provided for learning the weights associated with the median aggregation can be useful as alternative to the back propagation method.

References

[1] R.R Yager, Quantifier guided aggregation using OWA operators, International Journal of Intelligent Systems 11 (1996) 49–73.
[2] R.R. Yager, A. Rybalov, Understanding the median as a fusion operator, International Journal of General Systems 26 (1997) 239–263.
[3] R.R. Yager, On ordered weighted averaging aggregation operators in multi-criteria decision making, IEEE Transactions on Systems, Man and Cybernetics 18 (1988) 183–190.
[4] D. Dubois, H. Prade, A review of fuzzy sets aggregation connectives, Information Sciences 36 (1985) 85–121.
[5] R.R. Yager, On mean type aggregation, IEEE Transactions on Systems, Man and Cybernetics 26 (1996) 209–221.
[6] B. Widrow, R. Winter, Neural nets for adaptive filtering and adaptive pattern recognition, Computer 21 (1988) 25–39.
[7] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Transactions on Fuzzy Systems 4 (1996) 103–111.
[8] R.R. Yager, Fuzzy logic control with discrete outputs, in: Proceedings of World Congress on Neural Networks, Washington, DC, II:595–601 1995.